$w(t) = \begin{cases} 1 & 0 \le t \le T_f \\ 0 & , \text{o.w} \end{cases}$ & $g(t) = \beta \cos(2\pi f_c t)$

$h(t) = w(t) \cdot g(t)$

$h(t) = \begin{cases} \beta \cos(2\pi f_c t) & 0 \le t \le T_f \\ 0 \, , & \text{O.W} \end{cases}$

$w(t)$ is rect.

$W(jw) = T_f \, \text{sinc}\left(\frac{wT_f}{2\pi}\right)$ $\quad w = 0$ $\quad$ & $G(jw) = \frac{\beta}{2}[\delta(w - 2\pi f_c) + \delta(w + 2\pi f_c)$

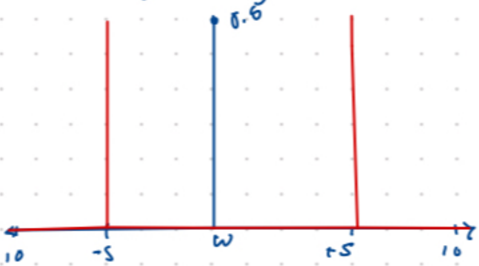
$w = \pm 2\pi f_c$

$\Rightarrow \quad H(jw) = W(jw) * G(jw)$ (convolution)

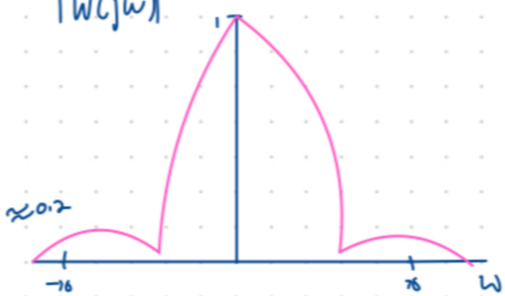
centered @ $w = \pm 2\pi f_c$

$$H(jw) = \frac{\beta T_f}{2} \text{sinc}\left(\frac{(w - 2\pi f_c)T_f}{2\pi}\right) + \frac{\beta T_f}{2} \text{sinc}\left(\frac{(w + 2\pi f_c)T_f}{2\pi}\right)$$

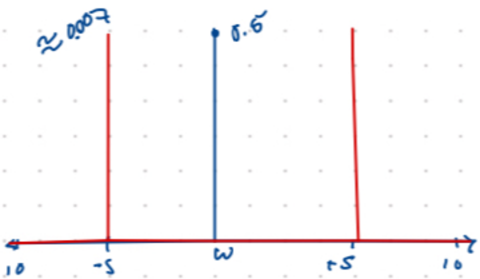
$|G(jw)|$



$|W(jw)|$



$|H(jw)|$



The sinc functions work as centered $\pm 2\pi f_c$.

Frequencies near pass through the filter

& frequencies far from decay from sinc function

```
%Melissa Regalado
%U29407369
%EC401 Lab 5
```

Lab Problem 5.1

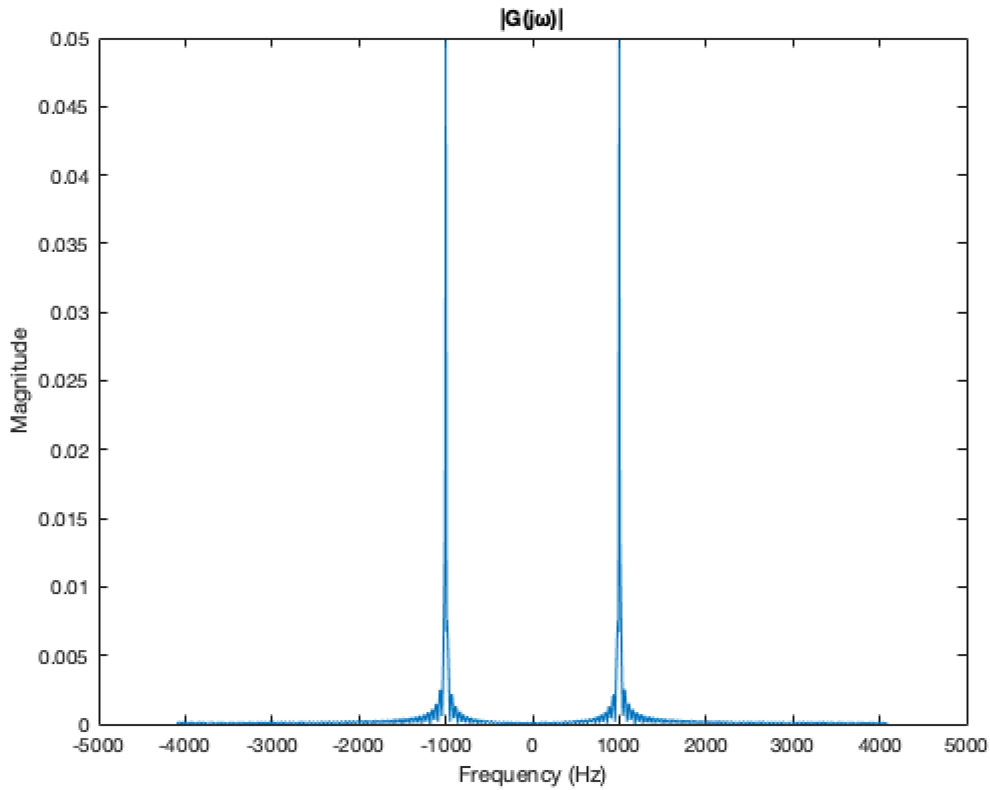
```
Fs = 8192;
fc = 1000;
Tf = 0.01;
t = 0:1/Fs:0.1;
B = 1;
g = B*cos(2*pi*fc*t);
w = (t<Tf);
h = w.*g;

cd('/Users/melissaregalado/Documents/MATLAB/EC401/Lab5/');

% compute FT
[G, f] = ctft(g, Fs, 1000);
[W, f] = ctft(w, Fs, 1000);
[H, f] = ctft(h, Fs, 1000);

%plot
figure(1);
plot(f, abs(G));
title('|G(jω)|');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

figure(2);
plot(f, abs(W));
title('|W(jω)|');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

figure(3);
plot(f, abs(H));
title('|H(jω)|');
xlabel('Frequency (Hz)');
ylabel('Magnitude');


%c.)

Fs = 8192;
fc = 1000;


% first filter duration
Tf1 = 0.005;
h1 = dtmffilters(fc, Tf1, Fs);
```

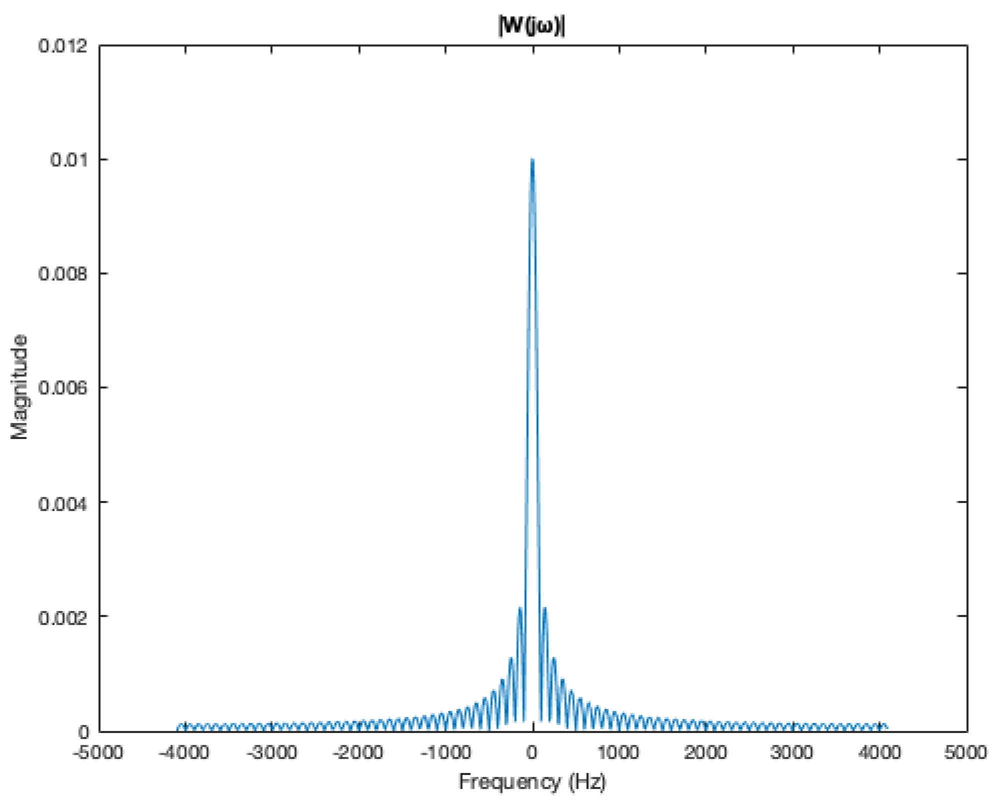
```
[H1, f] = ctft(h1, Fs, 10000);


figure(4);
plot(f, abs(H1));
title('|H(j\omega)| with Tf = 0.005');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
axis([0 2000 0 1]);

% sec filter duration
Tf2 = 0.008;
h2 = dtmffilters(fc, Tf2, Fs);
[H2, f] = ctft(h2, Fs, 10000);
figure(5);
plot(f, abs(H2));
title('|H(j\omega)| with Tf = 0.008');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
axis([0 2000 0 1]);

% thirdfilter duration
Tf3 = 0.02;
h3 = dtmffilters(fc, Tf3, Fs);
[H3, f] = ctft(h3, Fs, 10000);
figure(6);
plot(f, abs(H3));
title('|H(j\omega)| with Tf = 0.02');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
axis([0 2000 0 1]);

% four filter duration
Tf4 = 0.05;
h4 = dtmffilters(fc, Tf4, Fs);
[H4, f] = ctft(h4, Fs, 10000);
figure(7);
plot(f, abs(H4));
title('|H(j\omega)| with Tf = 0.05');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
axis([0 2000 0 1]);


%c.2)

%Larger TF indicates a more narrow passbamnd and definite
% stop bands that better neglect unwanted frequencies . Smaller
% TF allows more  freq leakage on edges

%d.1)
% uses a narrower passband to isolate freq while ones ooutside of
% range are attenuated.
```

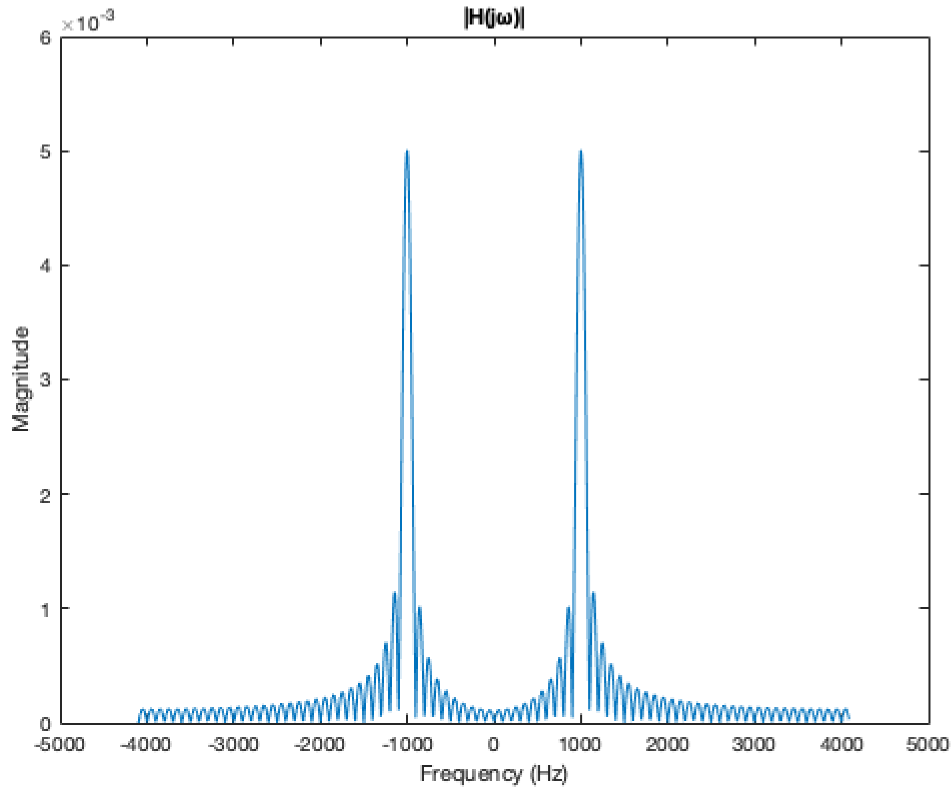
```
Fs = 8192;
Tf = 0.027;

% center frequencies
freq = [697, 770, 852, 941, 1209, 1336, 1477];
randomized_freq = freq + (rand(1, length(freq)) - 0.5) * 10;


for i = 1:length(freq)
    fc = freq(i);
    h = dtmffilters(fc, Tf, Fs);       %  bandpass filter
    [H, f] = ctft(h, Fs, 10000);       % FT

    figure(i + 7);
    plot(f, abs(H));
    title(['Frequency Response of Bandpass Filter at ', num2str(fc), ' Hz']);
    xlabel('Frequency (Hz)');
    ylabel('Magnitude');
    axis([0 2000 0 1]);
end

%%Lab Problem 5.2

Fs = 8192;
Tf = 1.0;
frequencies = [697, 770, 852, 941, 1209, 1336, 1477];
key = '5';
sKey = dtmfdial(key, 1.3, Fs);

for i = 1:length(frequencies)
    fc = frequencies(i);
    h = dtmffilters(fc, Tf, Fs);
    [score, E] = dtmfdetect(sKey, h, Fs);
    fprintf('Frequency %d Hz | Score: %d | Energy: %.4f\n', fc, score, E);
end

%2.b)

Fs = 8192; % Sample rate
Ttone = 0.1; % Tone duration
KeysIn = '555-1212'; % Input key sequence

sKeys = dtmfdial(KeysIn,Ttone,Fs); % Make DTMF signal
sound(sKeys,Fs) % Play the signal

Tf_values = [.1];
for Tf = Tf_values
    fprintf('Testing with Tf = %.4f seconds\n', Tf);
    KeysOut = dtmfkeys(sKeys, Tf, Fs);
    fprintf('Decoded Key Sequence: %s\n', KeysOut);
end
```

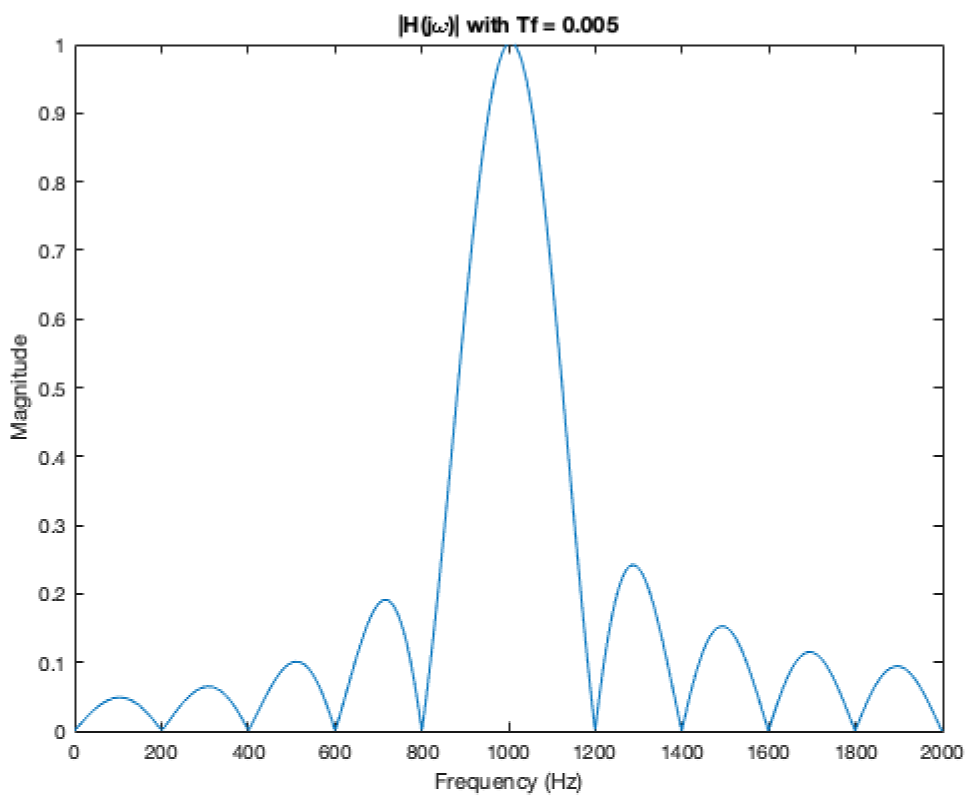
```
gain = 0.2;
noise = gain * (rand(1, length(sKeys)) - 0.5);
noisyKeys = sKeys + noise;

%  noisy DTMF signal
KeysOut_noisy = dtmfkeys(noisyKeys, Tf, Fs);

fprintf('Decoded Key Sequence with Noise: %s\n', KeysOut_noisy);


Tf_short = 0.05;
KeysOut_shortTf = dtmfkeys(sKeys, Tf_short, Fs);

fprintf('Decoded Key Sequence with Shorter Tf: %s\n', KeysOut_shortTf);


% Another approach can FT that can identify specific peaks
%while ignoring freq that dont relate. This would be easier to distinguish
%from noise and interference.
```

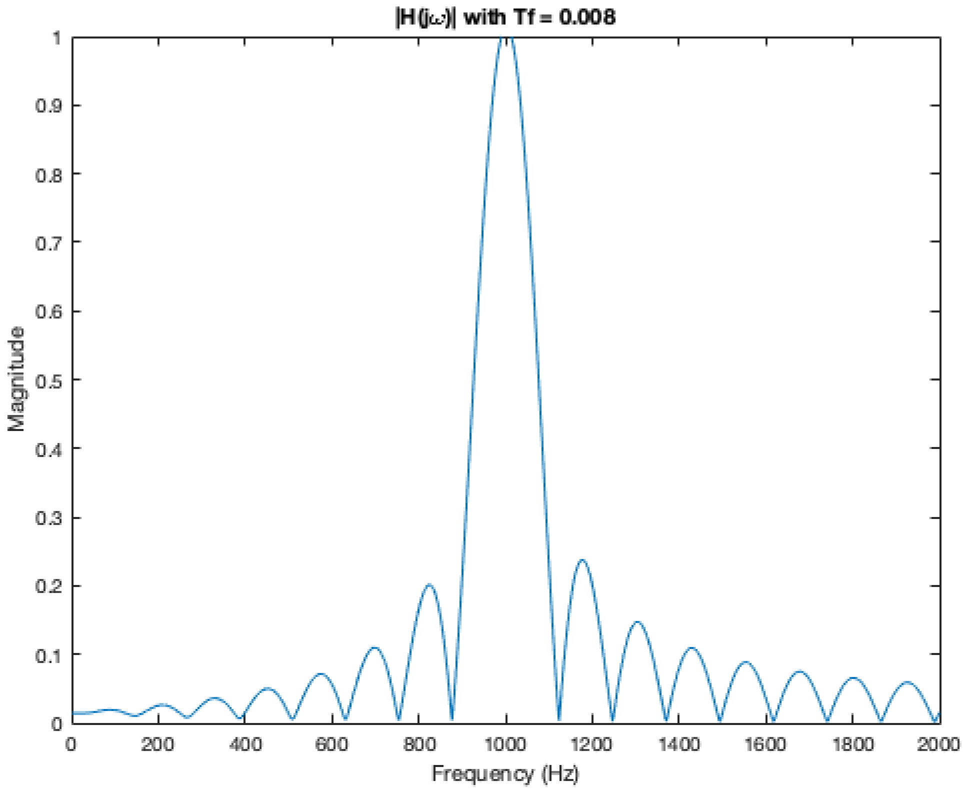
```
Frequency 697 Hz | Score: 0 | Energy: 0.0000
Frequency 770 Hz | Score: 1 | Energy: 0.7162
Frequency 852 Hz | Score: 0 | Energy: 0.0000
Frequency 941 Hz | Score: 0 | Energy: 0.0000
Frequency 1209 Hz | Score: 0 | Energy: 0.0000
Frequency 1336 Hz | Score: 1 | Energy: 0.7162
Frequency 1477 Hz | Score: 0 | Energy: 0.0000
Testing with Tf = 0.1000 seconds
Decoded Key Sequence: 5551212
Decoded Key Sequence with Noise: ?
Decoded Key Sequence with Shorter Tf: 5551212
```

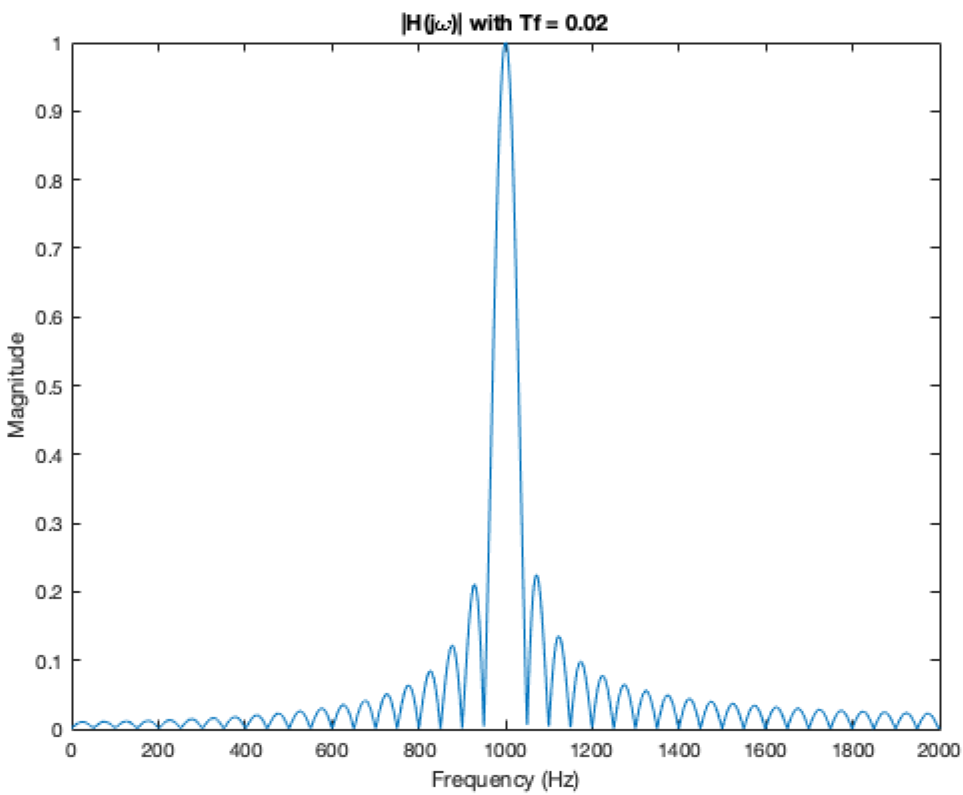

|G(jω)|
Magnitude
Frequency (Hz)
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000

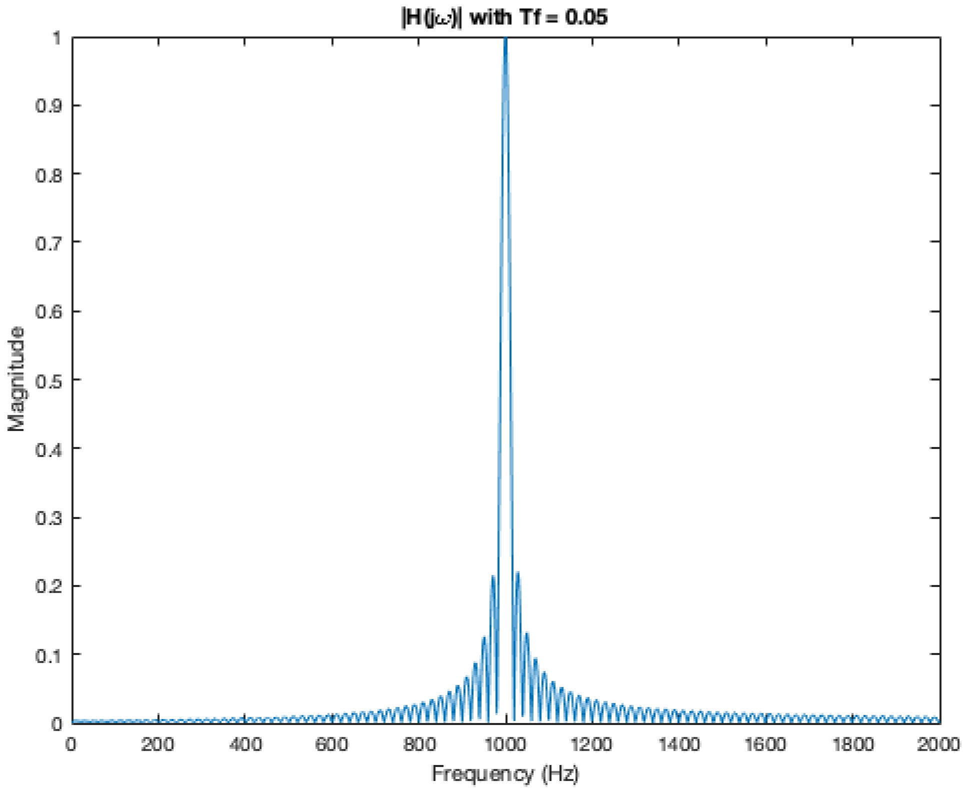

|W(jω)|
Magnitude
Frequency (Hz)
0.012
0.01
0.008
0.006
0.004
0.002
0
-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000

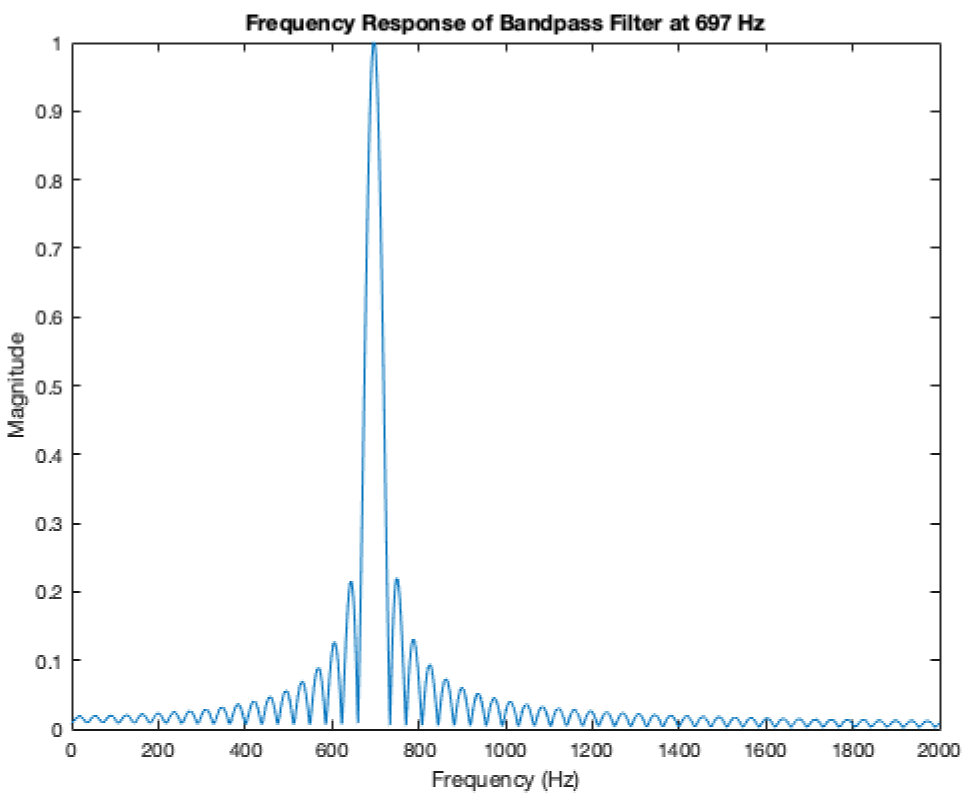

|H(jω)|
×10^-3
Magnitude
Frequency (Hz)
6
5
4
3
2
1
0
-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000

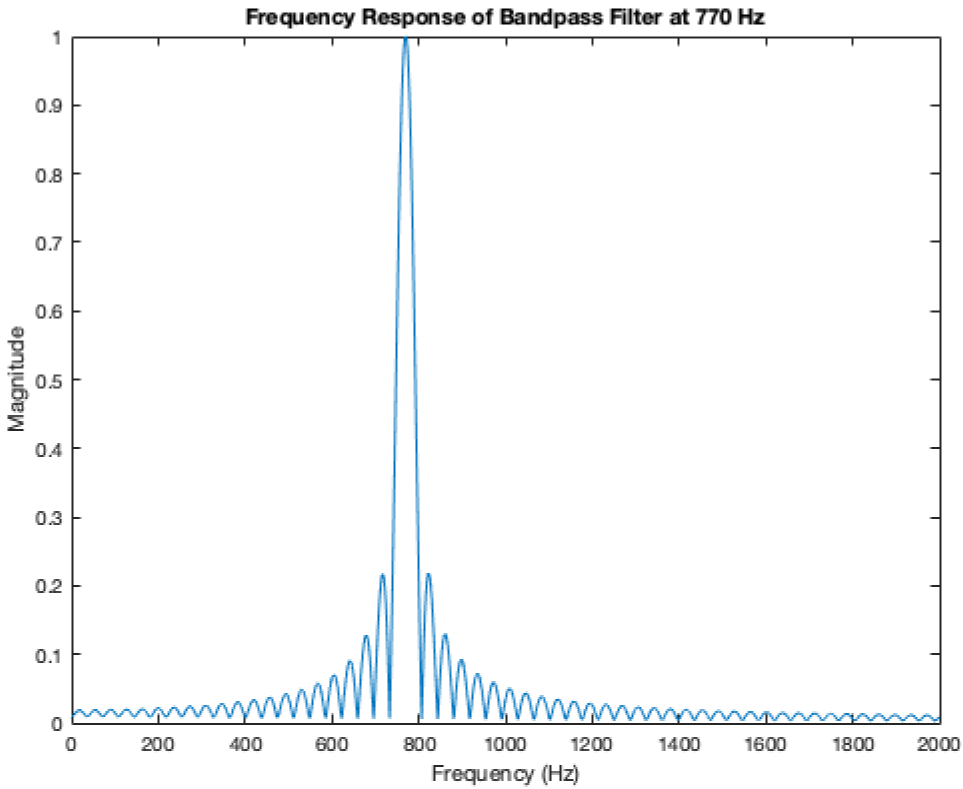

|H(jω)| with Tf = 0.005
Magnitude
Frequency (Hz)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000

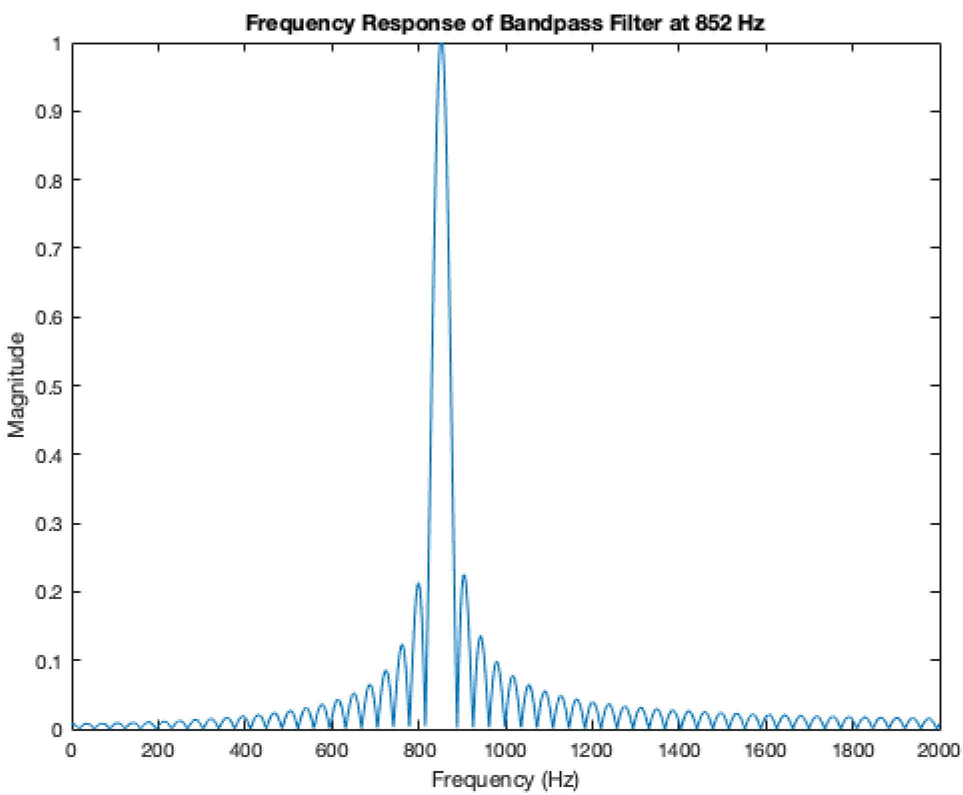

|H(jω)| with Tf = 0.008
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

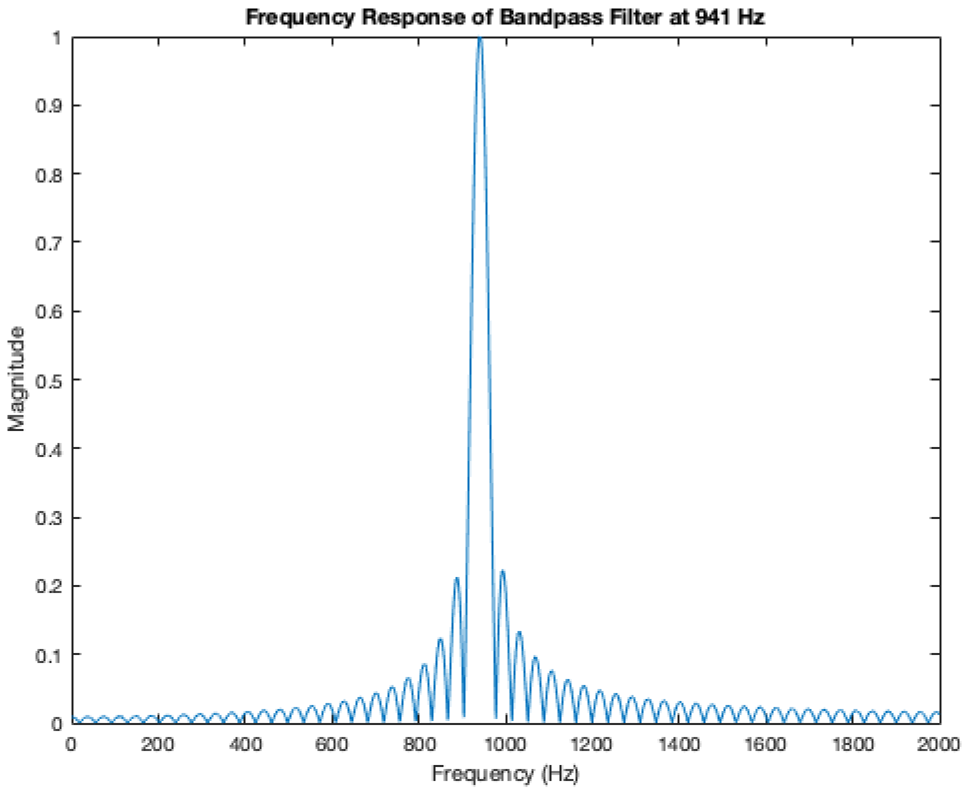

|H(jω)| with Tf = 0.02
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

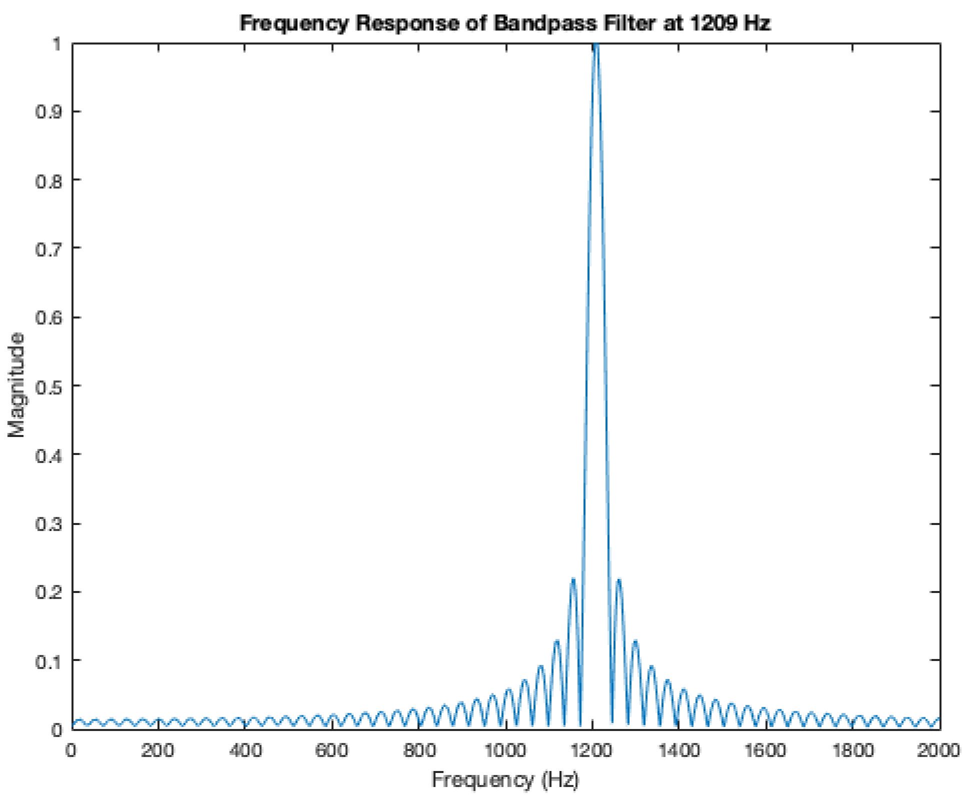

|H(jω)| with Tf = 0.05
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

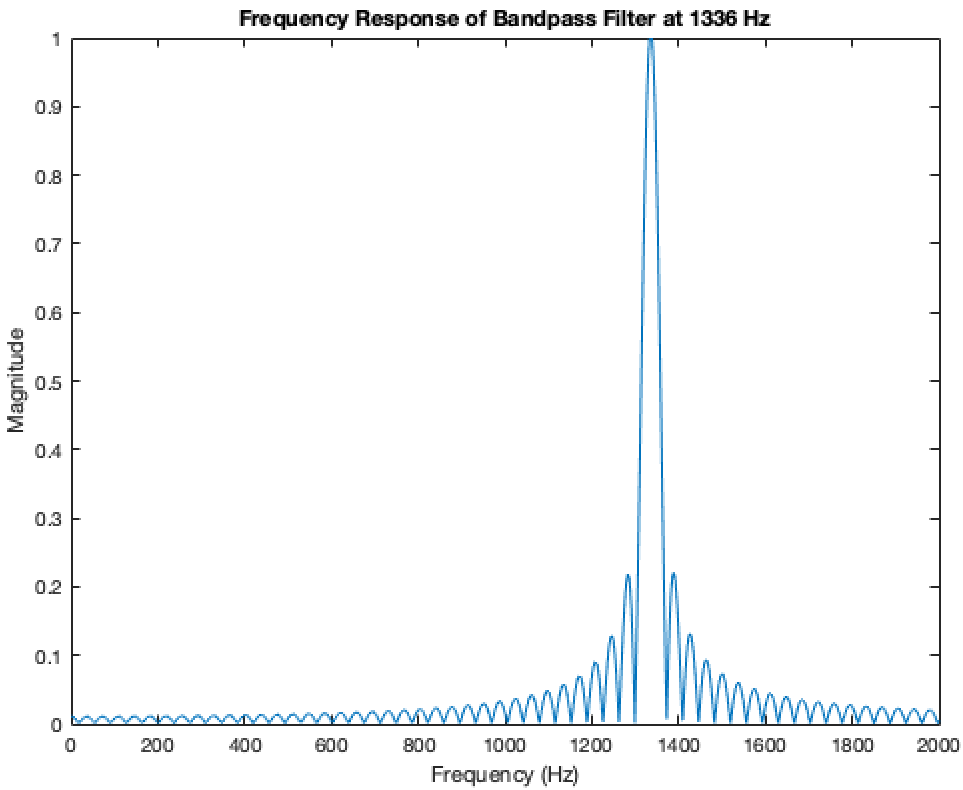

Frequency Response of Bandpass Filter at 697 Hz
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Frequency Response of Bandpass Filter at 770 Hz
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

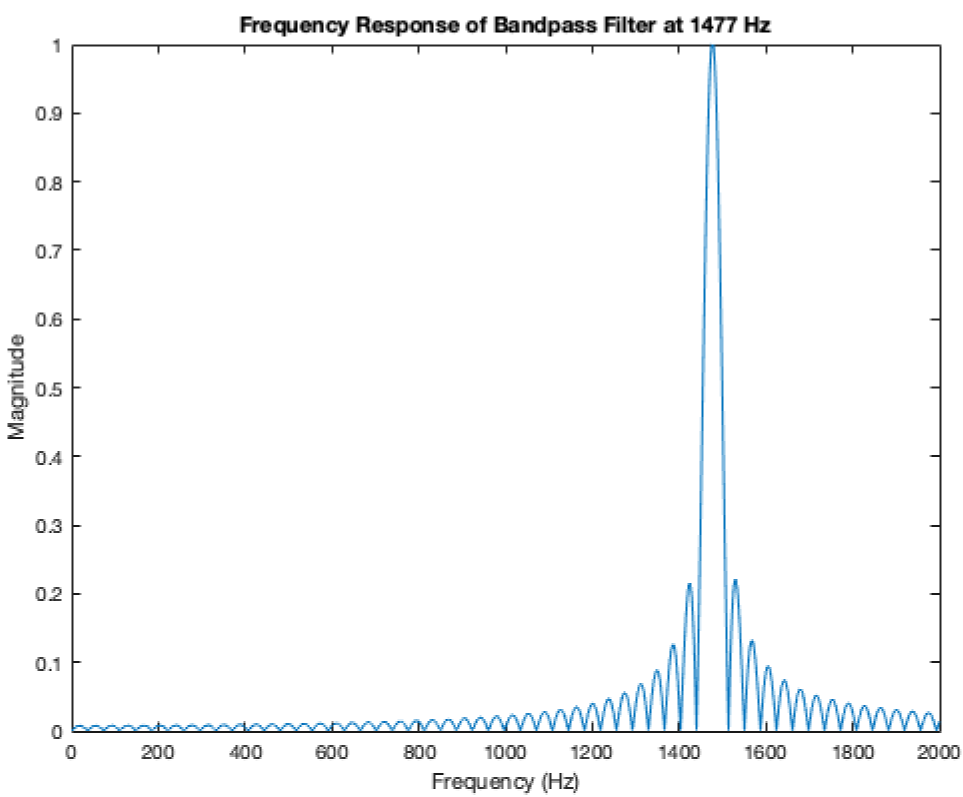

Frequency Response of Bandpass Filter at 852 Hz
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Frequency Response of Bandpass Filter at 941 Hz
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000


Frequency Response of Bandpass Filter at 1209 Hz
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000



Frequency Response of Bandpass Filter at 1336 Hz
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000



Frequency Response of Bandpass Filter at 1477 Hz
Magnitude
Frequency (Hz)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
200
400
600
800
1000
1200
1400
1600
1800
2000

Published with MATLAB® R2024b